DESIGN AND BUILD AN EVTOL DRONE

By

Chenyang Huang

Hongfan Liu

Xuan Chen

Zhengpu Ye

Final Report for ECE 445, Senior Design, [Spring 2024]

TA: Yanzhao Gong

6 May 2024

Project No. 42

Abstract

We have designed a five axis vertical takeoff and landing drone through mechanical structure and circuit control. Compared to ordinary drones, we have achieved greater payload and longer flight distance as much as possible by changing the structure and circuit operation of the drones. And on this basis, we have added visual functions to the drone. We can watch the images transmitted from the drone's perspective through online live streaming and perform car recognition on the images.

目录

1. Introduction

In recent years, the load-bearing transport UAV technology has been rapidly developed, and its practical applications in agriculture, medical rescue, logistics distribution, environmental monitoring and other fields are increasingly extensive, becoming an important force to promote social progress and technological innovation

In addition, transport drones also have potential applications in commercial photography, geological exploration, rapid building construction and other industries.

As technology continues to advance, the endurance, reliability and intelligence of drones are increasing, enabling them to carry out more complex missions and make a difference in a wider range of fields.

It can be said that heavy-carrying transport drones are gradually becoming the "air workers" of modern society, showing great potential and value in improving work efficiency, responding to public health events, ensuring supply chain stability, and promoting economic development. With the improvement of regulations and systems and the improvement of public acceptance, the application prospects of transport drones will be broader.

1.1 Section Head

In order to better coordinate our work and achieve interdisciplinary collaboration, we divided the work content from hardware to software into three parts: structural design, electronic control and visual design, and divided their functions into different architectures.

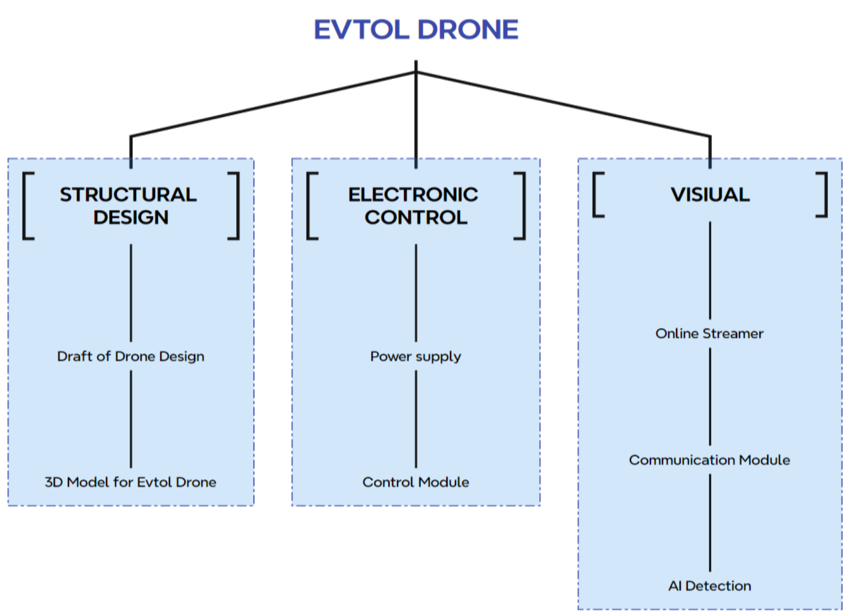

figure1. section head

1.2 High-level Requirements

The plane must be able to fly properly with load.

Electric control is to ensure hover and control.

Stable communication in 10 km and 30 minutes, and upload the camera image to the cloud.

2. Design

Our block diagram is shown below.

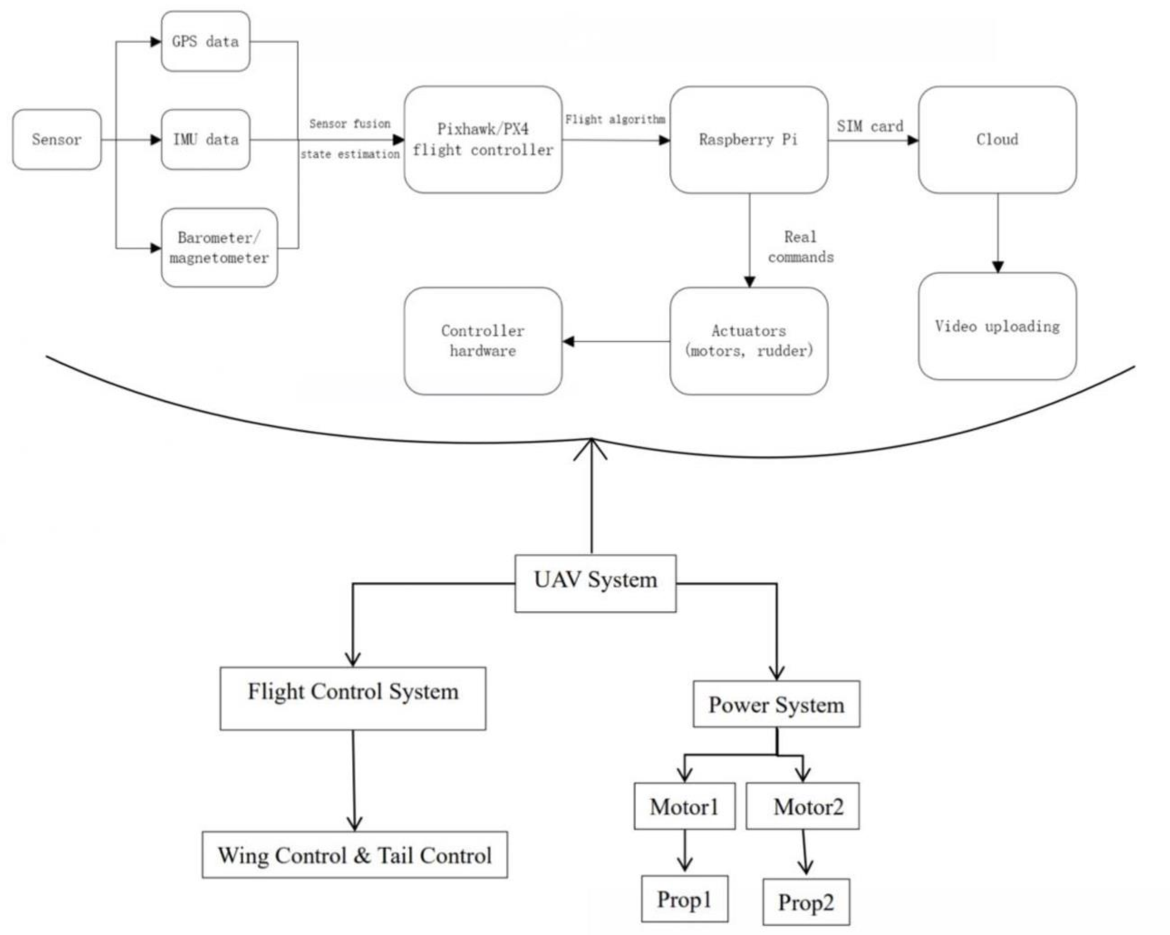

figure2. block diagram

2.1 Structural Design

The structural design part mainly includes the shape of the airplane, the position of the motor, the overall fixing as well as the assembly.

Our design phase is divided into three main stages: early idea, detail design, and adjustment and improvement. This section will be specified next.

2.1.1 Early Idea

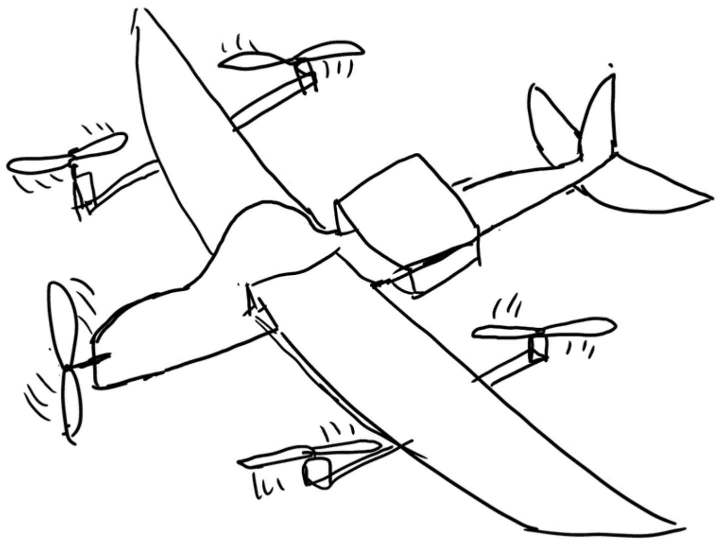

Figure3. Draft of Drone Design

In the beginning, considering the function of vertical take-off and landing, then it is necessary to have devices that can provide direct upward force. At the same time, it is necessary to realize long time horizontal flight, then it is necessary to have fixed wings and forward propulsion power. Referring to existing UAV structures, Figure 1 shows our initial design.

The fuselage is made of a lighter epo material, four axes are used to provide vertical lift, and a propulsion motor and fixed wing are used to ensure horizontal flight. This is what we envisioned at the beginning.

2.1.2 Details Design

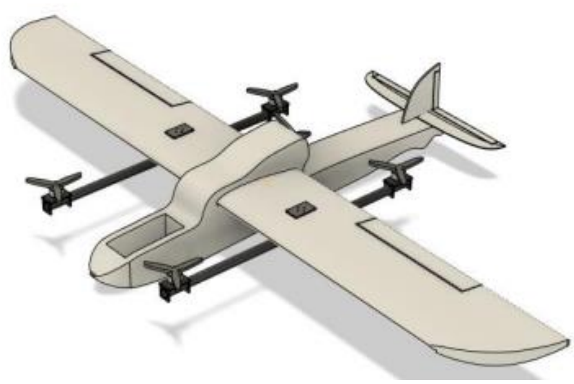

Figure4. 3D Model for Evtol Drone

After having an initial idea, the specific structure was first designed in 3d modeling software. At this stage, the specific connections between the 4-axis motors and the carbon tubes, as well as between the carbon tubes and the fixed wing, were addressed. We mainly used bolts and nuts to fix them, and also used hexagonal aluminum posts as well as some printed parts.

During this period, by evaluating the weight of the plane, the right motors as well as batteries were chosen, while the carbon tubes were customized and the fixings were printed with the school's 3d printer.

2.1.3 Assembly, adjustment and refinement

After simply building and completing the airplane, many problems were discovered.

The first is that the 4-axis motors are not securely fastened and can wobble back and forth. This is caused by the softer wings. For this reason, we added two more carbon tubes to make the structure more stable as Figure 3 shows.

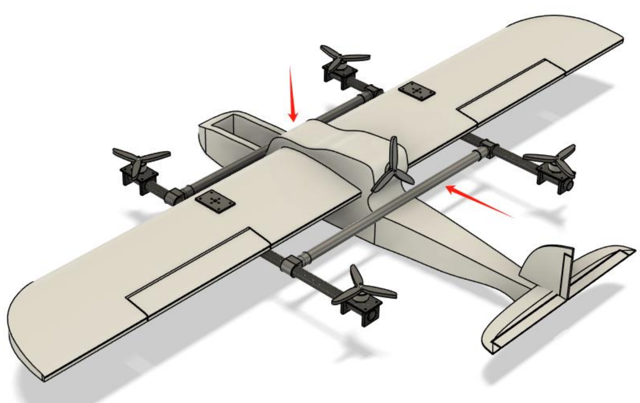

Figure5. New carbon tubes

Secondly,we changed the model of the motor as we found the original motor to be underpowered. At the same time, the new motors are connected with XT60 interface, which reduces the problems caused by soldering.

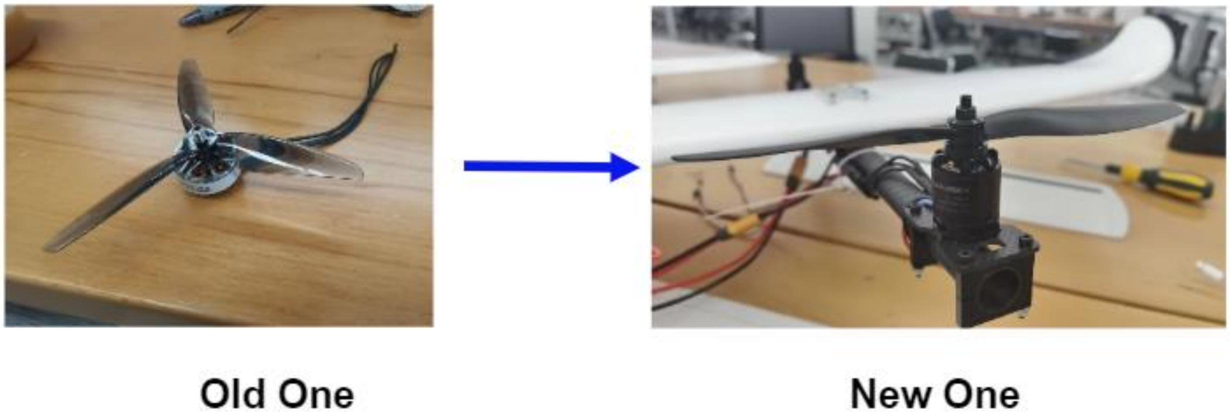

Figure6. Change of Motor

We also found that the center of gravity of the plane was forward, so we moved the battery position from the original nose to the center of the fuselage. There is also the problem of fixing the position of the circuit components. Originally we planned to put everything in the nose cavity. But in reality this would have the problem of making the center of gravity unstable, as well as inconvenient debugging and lack of space. So we fixed the ESCs to the new carbon tubes that are horizontal, and put the splitter plate on the fuselage. The pixhawk is fixed to the outside of the headstock, so that it is easy to adjust.

After the above optimizations, Figure 5 below shows our final version.

Figure7. Actual product

2.2 Power supply

The circuit is powered by a BOSLI-PO battery with a 1300mAh capacity. Our power budget is 195A (maximum) at 22.2V, mostly consumed by five propeller motors. Each motor consumes 36.6A of current. This battery is chosen for its affordability and sufficient power.

Figure 8 BOSLI-PO battery

TABLE 1 Requirement and Verification of Power supply

Requirement	Verification
Outputs larger than 183A (36.6A*5=183A) at 22.2V	A. Measure the open-circuit voltage with a voltmeter, ensuring that it is at 22.2V B. Ensure that the current through the motors are about 36.6A using an ammeter in series

2.3 Control Module

The control module receives signal from drone controller and controls the flight. It contains a Pixhawk flight controller, a remote-control receiver, a remote control and five electronic speed controllers.

2.3.1 Pixhawk flight controller

The Pixhawk 2.4.8 Drone Flight Controller works as the brain of our drone. It analyzes the commands of the user and transmitting electrical signals to electronic speed controllers. We choose this version because of its powerful functions and friendly price.

Figure9. Pixhawk flight controller

TABLE 2 Requirement and Verification of Pixhawk flight controller

Requirement	Verification
Analyze commands and regulate the motors.	Works successfully in Mission Planner ground control.

2.3.2 Micro zone MC6C remote control and MC7RB remote-control receiver

The remote control and receiver make sure that the flight could receive our commands. It works will in our computer software simulation

Figure10. Micro zone MC6C remote control and MC7RB remote-control receiver

2.3.3 BLHe li-S electronic speed controllers

The electronic speed controllers (ESCs) control the speed of motors. BLHe li-S ECSs are chosen for compatibility of protocols.

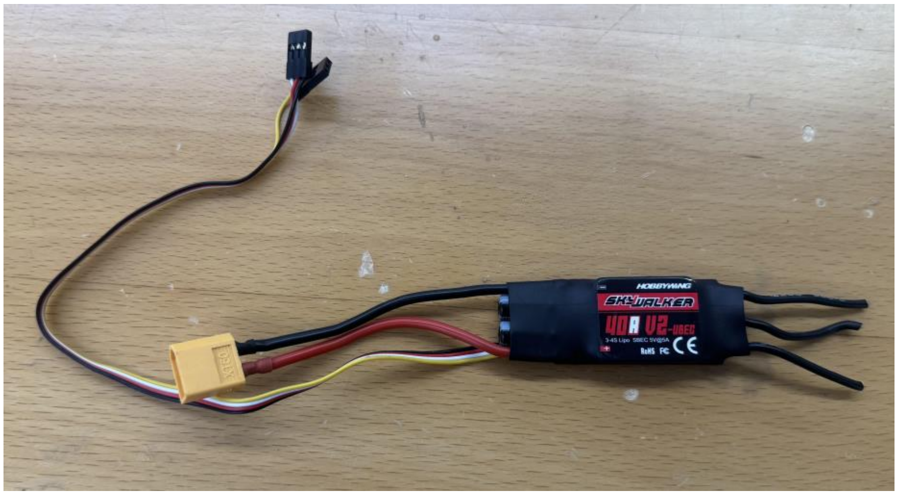

Figure11. BLHe li-S electronic speed controller

2.3.4 Ground Control Software

We choose Mission Planner Vtol survey as our Ground Control. It is a ground control station for Plane, Copter and Rover. It is compatible with Windows only. Mission Planner can be used as a configuration utility or as a dynamic control supplement for your autonomous vehicle.[1]

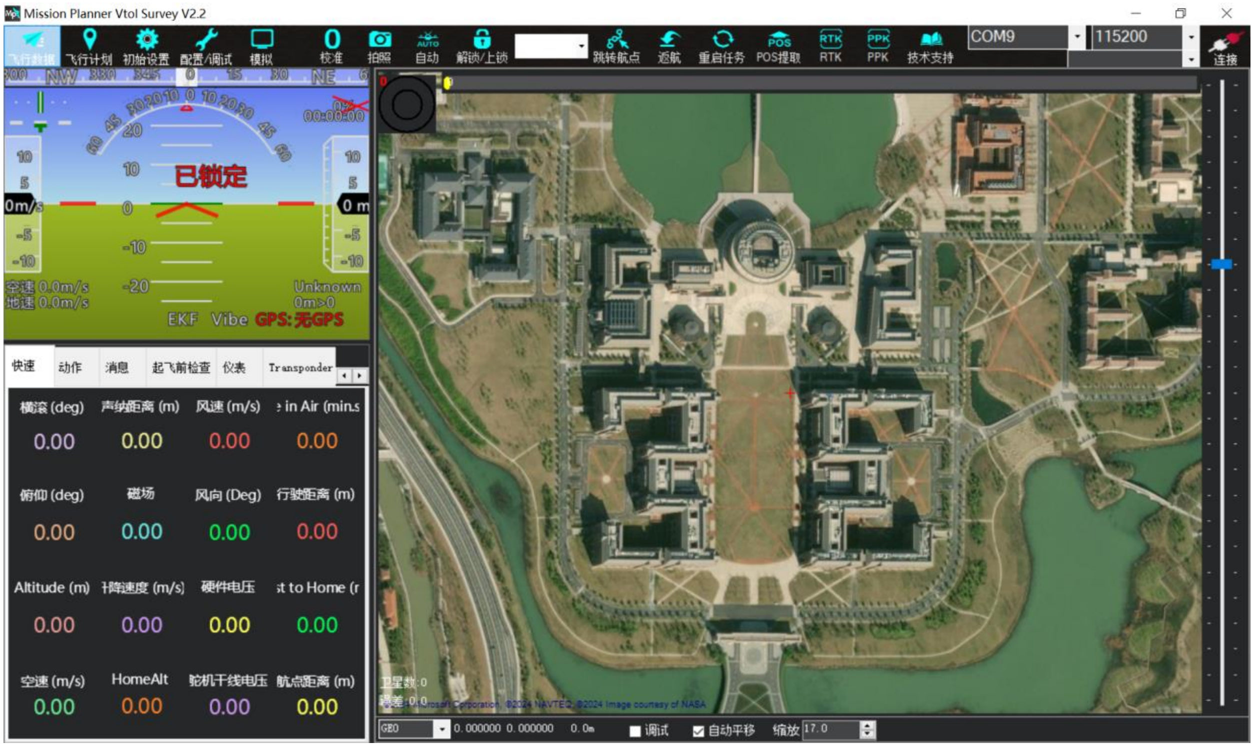

Figure12. Mission Planner Vtol survey

2.4 visual functions

In our drone construction process, we also added visual functions to the drone. This function mainly consists of a Raspberry Pi PCB board, a USB camera, and a communication module based on a SIM card. Below, we will elaborate on the following points.

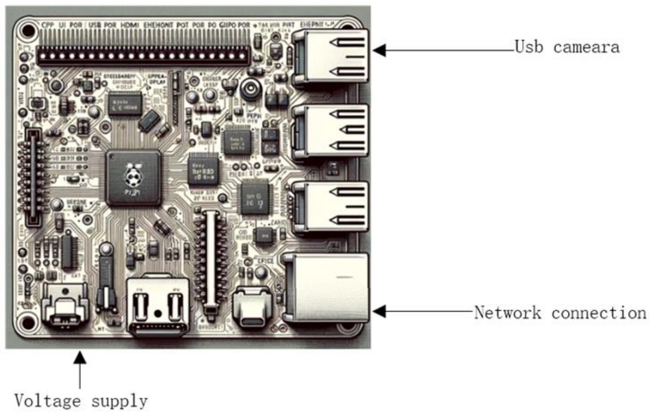

figure13. Rasperry Pi

2.4.1 Rasperry Pi(3B) in evtol drone

The Raspberry PI has five main functions as below:

Controller Raspberry pie can serve as the main controller of drones, responsible for processing flight control logic and responding to remote control commands. It can be used in conjunction with flight control boards (such as Pixhawk) to achieve more complex flight modes and automatic navigation functions.

Real time video transmission Using its USB interface, Raspberry Pi can connect to a USB camera, capture real-time video, and transmit it back to the ground station via wireless network, providing the operator with a first person view (FPV) or video monitoring.

Data collection and processing Raspberry pie can connect multiple sensors (such as GPS, barometer, temperature sensor, etc.) to collect various data during flight. These data can be used for flight log recording, environmental monitoring, geographic information system (GIS) data collection, etc.

Artificial Intelligence and Image Processing Raspberry pie has the ability to process images and run lightweight artificial intelligence models. This can be used for real-time target recognition, tracking, and obstacle avoidance, enhancing the autonomous flight capability of drones.

Communication and network functions Raspberry Pi supports Wi Fi and Bluetooth communication and can be used for remote control and data transmission of drones. In addition, Raspberry Pi can also achieve network communication with other drones or ground stations, supporting complex collaborative tasks and group flights

2.4.2 USB connected micro camera

Raspberry pie combined with a USB camera can build a low-cost and powerful video surveillance system. Using software like Motion can achieve motion detection, video recording, and real-time streaming functions. In the project, USB cameras can serve as the "eyes" of robots, helping them navigate, avoid obstacles, and recognize specific objects or signs. This is particularly important for the development of autonomous guided vehicles, drones, or other types of autonomous robots. Our drone will primarily use the mjpg-streamer module for video communication and OpenCV Python code training for deep learning, ultimately achieving object recognition functionality.

figure14. usb-camera

figure15. 4G communication module

2.4.3 SIM card communication module

After possessing the above hardware, our visual system can now broadcast videos on the same LAN, but we still need a self-built communication module to simulate communication functions in real environments.

We choose to independently build a communication module based on SIM cards. The main function of connecting SIM cards to Raspberry Pi to build a network module in communication is to provide mobile network access capability for Raspberry Pi. This setting can expand the functionality of Raspberry Pi, allowing for data transmission and communication without traditional wired networks such as Ethernet or Wi Fi. The most important functions are data transmission and Remote control. Raspberry pie can send and receive data through mobile networks, which is very useful for remote monitoring and management systems. It can remotely log in to Raspberry Pi, execute commands, or update configurations from anywhere in the world through a mobile network.

In addition, having a SIM card communication module can even provide communication and data services. For example, SMS service: The SIM card module allows Raspberry Pi to send and receive SMS, providing another form of communication for certain applications. Voice call: Some SIM card modules also support voice calls, and Raspberry Pi can be used as part of the phone system.

The raspberry pi can even act like a phone.

2.4.4 Implementation of video live streaming

Setting up a camera for live streaming on Raspberry Pi can be achieved through various methods, and our approach is to use mjpg streamer. Since mjpg streamer is not in the standard repository of Raspberry Pi, we will compile it from the source code.

After compiling and installing mjpg streamer, it can be run to start the video stream, and the video stream can be obtained by agreeing to access the Raspberry Pi IP through the still and configured access on the local area network.

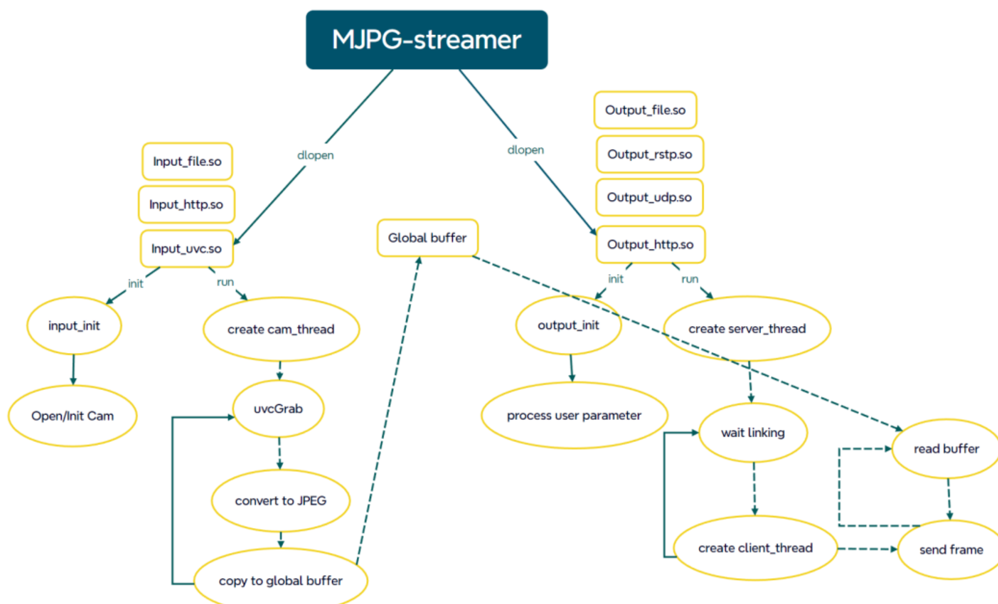

figure16. mjpg-streamer function

As a lightweight video streaming service software, mjpg streamer has the following advantages:

Low latency: mjpg streamer streaming servers have lower latency during the streaming process, which is crucial for applications that require fast response, such as remote control or monitoring.

Lightweight: As it is written in C language, it is relatively lightweight, occupies less system resources, and is suitable for running on resource limited devices, such as Raspberry Pi.

High compatibility: Supports multiple types of USB cameras and operating systems, which means you can use it on different hardware and software configurations.

2.4.5 Rental and video uploading of cloud platforms

In this step, we will first rent a cloud platform, such as Alibaba Cloud or Huawei Cloud. Then configure video capture and encoding settings. For example, we will use FFmpeg to encode the video stream into a format suitable for Internet transmission. Configure the transmission of video streams according to the requirements of the cloud platform.

This may involve setting specific protocols (such as RTMP or HLS) and the URL of the flow destination. Then start video streaming and use FFmpeg or other tools to capture the video and transfer it to the cloud platform. The final step also involves processing and distribution on the cloud platform. This may include video analysis, storage, and transcoding. We will configure access control and security as needed to ensure the secure transmission and access of video streams.

figure17. cloud server

figure19. CNNAME configuration and status

2.4.6 Use Yolov5 to complete image recognition

By integrating the deep learning algorithm based on YOLOv5 (You Only Look Once version 5) and UA-DETRAC vehicle detection data set, we realized real-time monitoring and identification of highway vehicles. The system continuously captures highway video streams while in flight and analyzes road conditions to provide critical information for traffic management, safety supervision and accident response.

YOLOv5 is one of the leading target detection frameworks and is known for its excellent detection speed and accuracy. In our project, the YOLOv5 model has been specially trained to adapt to the characteristics of various types of vehicles in the highway environment. In addition, the UA-DETRAC public data set, which contains diverse vehicle images and annotation information, further enhances the model's ability to detect vehicles of different types, sizes and in complex traffic scenarios.

The trained model is deployed on the drone, the camera equipment on board transmits the image back to the server in real time, and the vehicle recognition model quickly processes the video data to detect and mark the location and trajectory of the vehicle.

```python
def parse_opt(known=False):
    data = "engine/configs/voc_local.yaml"
    workers = 8
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', type=str, default='engine/pretrained/yolov5s.pt', help='initial weights path')
    parser.add_argument('--cfg', type=str, default="yolov5s.yaml", help='model.yaml path')
    parser.add_argument('--data', type=str, default=data, help='dataset.yaml path')
    parser.add_argument('--hyp', type=str, default='data/hyps/hyp.scratch-v1.yaml', help='hyperparameters path')
    parser.add_argument('--epochs', type=int, default=300)
    parser.add_argument('--batch-size', type=int, default=8, help='total batch size for all GPUs')
    parser.add_argument('--imgsz', '--img', '--img-size', type=int, default=640, help='train, val image size (pixels)')
    parser.add_argument('--rect', action='store_true', help='rectangular training')
    parser.add_argument('--resume', nargs='?', const=True, default=False, help='resume most recent training')
    parser.add_argument('--nosave', action='store_true', help='only save final checkpoint')
    parser.add_argument('--noval', action='store_true', help='only validate final epoch')
    parser.add_argument('--noautoanchor', action='store_true', help='disable autoanchor check')
    parser.add_argument('--evolve', type=int, nargs='?', const=300, help='evolve hyperparameters for x generations')
    parser.add_argument('--bucket', type=str, default='', help='gsutil bucket')
    parser.add_argument('--cache', type=str, nargs='?', const='ram', help='--cache images in "ram" (default) or "disk"')
    parser.add_argument('--image-weights', action='store_true', help='use weighted image selection for training')
    parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
    parser.add_argument('--multi-scale', action='store_true', help='vary img-size +/- 50%%')
    parser.add_argument('--single-cls', action='store_true', help='train multi-class data as single-class')
    parser.add_argument('--adam', action='store_true', help='use torch.optim.Adam() optimizer')
    parser.add_argument('--sync-bn', action='store_true', help='use SyncBatchNorm, only available in DDP mode')
    parser.add_argument('--workers', type=int, default=workers, help='maximum number of dataloader workers')
```

figure20. yolo train.py

3. Design Verification

The primary objective of the Design Verification process for our eVTOL drone was to ensure that all design specifications were met, including lift capacity, stability during flight, energy efficiency, and safety standards. This was critical to guarantee functionality and compliance with aviation regulations.

3.1 Structural Function

3.1.1 Structural strength Test

Carbon tubes were inserted into the wing for added strength, while four carbon tubes were used to secure the motor position while keeping the motor level. After testing, the overall structure is able to maintain stability during flight.

3.1.2 Lift Capactiry Test

We measured the maximum lift capacity of the drone to ensure it could ascend while carrying the intended payload. The test was conducted in a outdoor environment, similiar to the real working environment. The lift capacity were within the expected thresholds, validating the aerodynamic design and motor configuration.

3.2 Flight Function

3.2.1 Flight Stability Test

Flight tests were performed to assess the drone’ s stability during takeoff, hovering, maneuvering, and landing. Data was collected using onboard sensors.

During the stability tests, the drone maintained a steady hover with minimal deviation and responded well to control inputs during maneuvers. There were oscillations during rapid directional changes which were noted for further investigation. The observed oscillations during flight maneuvers will require adjustments in the drone’ s flight control algorithms to enhance stability.

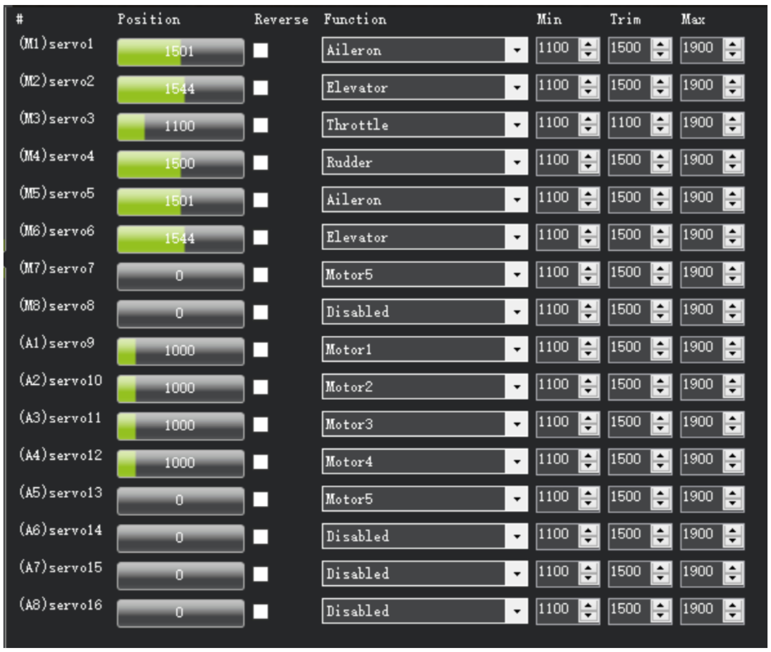

figure21. flight stability

3.2.2 Battary Efficiency Test

We conducted multiple flight cycles to determine the actual energy consumption versus the estimated consumption to verify battery efficiency under various operating conditions. The drone achieved an maximum flight time of over 20 minutes. The efficiency was within the acceptable range but indicated room for optimization.

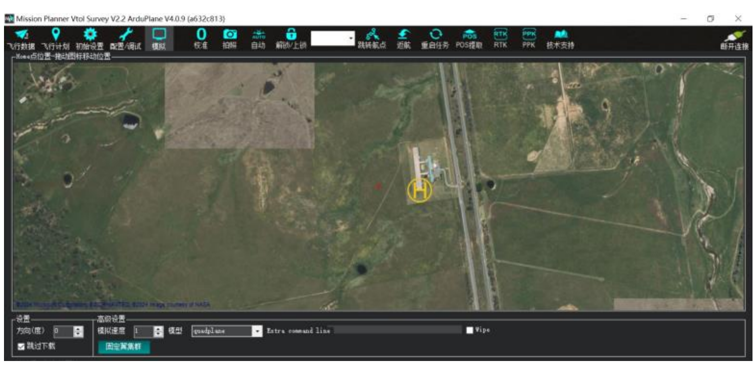

figure22. Battary Efficiency

3.3 Visual Functions

The ultimate goal of the vision part is to install a vision module based on Raspberry PI on the aircraft. To verify the integrity of the whole scheme, we need to ensure the stability of the Raspberry PI system, the stability of the camera capture picture, the stability of the uploaded live stream, the stability of 4G signal and the correctness of yolo recognition.

3.3.1 Rasperry Pi(3B) in evtol drone

In our above design, we used a Raspberry Pi as the core chip of the visual system. We first burned the Raspberry Pi operating system for Raspberry Pi. From the simple display screen below, you can see that Raspberry Pi can boot smoothly after being powered on.

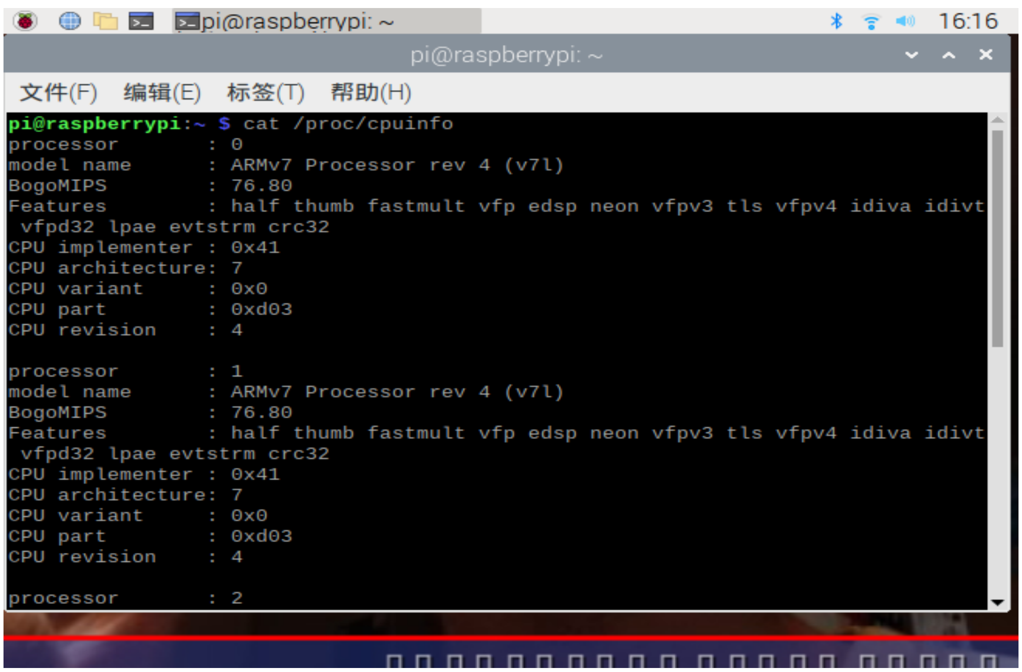

figure23. Rasperry Pi os info

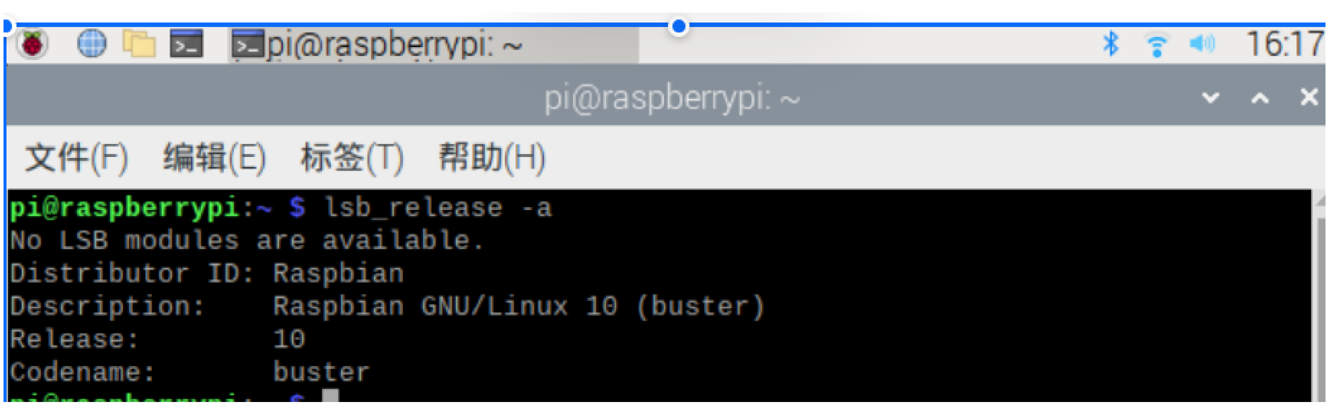

figure24. Rasperry Pi Raspbian info

3.3.2 External camera and mjpg live streaming module

By lsusb command we can see the usb connection between our raspberry pi and camera.

For mjpg-streaming module, the basic function of this module, as introduced earlier, is to synchronize and live stream the images captured by the camera to the same local area network. So let's first check the IP address of the Raspberry Pi in the current local area network.

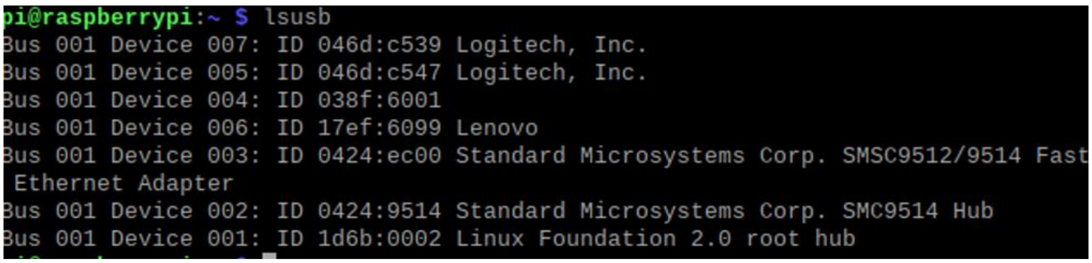

figure25. usb connection

figure26. ip address

Then, when we connect to the same local area network, we can open the corresponding IP address to receive the live video.

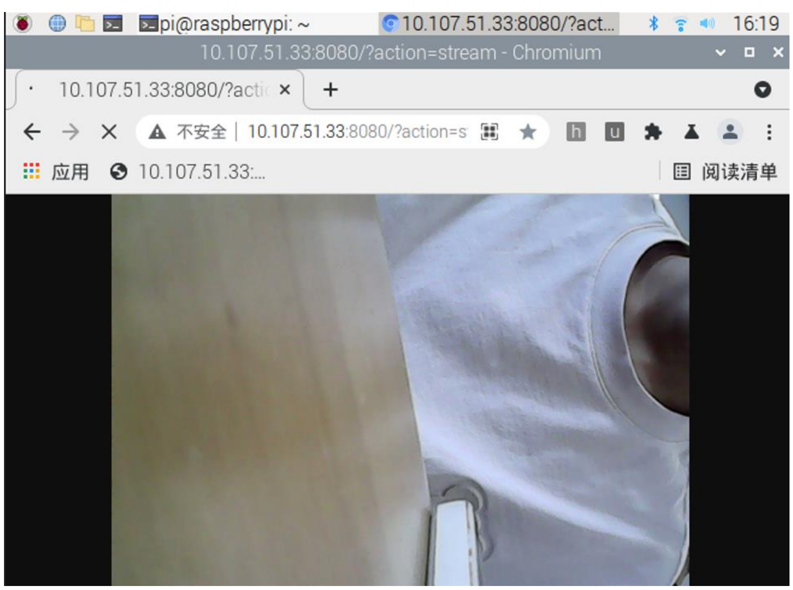

figure27. mjpg Live streaming within the local area network

3.3.3 Rental and video uploading of cloud platforms

We rented the OSS platform for this stage. we receive Raspberry Pi streams through local computers and push them again through the OSS platform to achieve full platform video live streaming.

In the end, we achieved this feature and were able to access live streaming across the entire network on any live streaming platform, rather than being limited to a local area network.

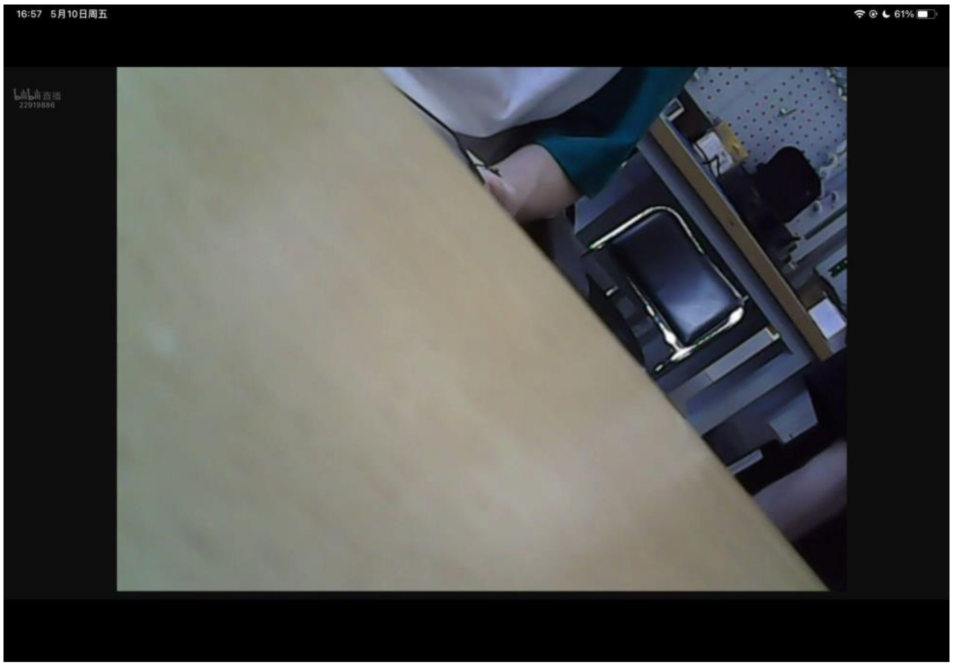

figure28. bilibili live (Non LAN connection)

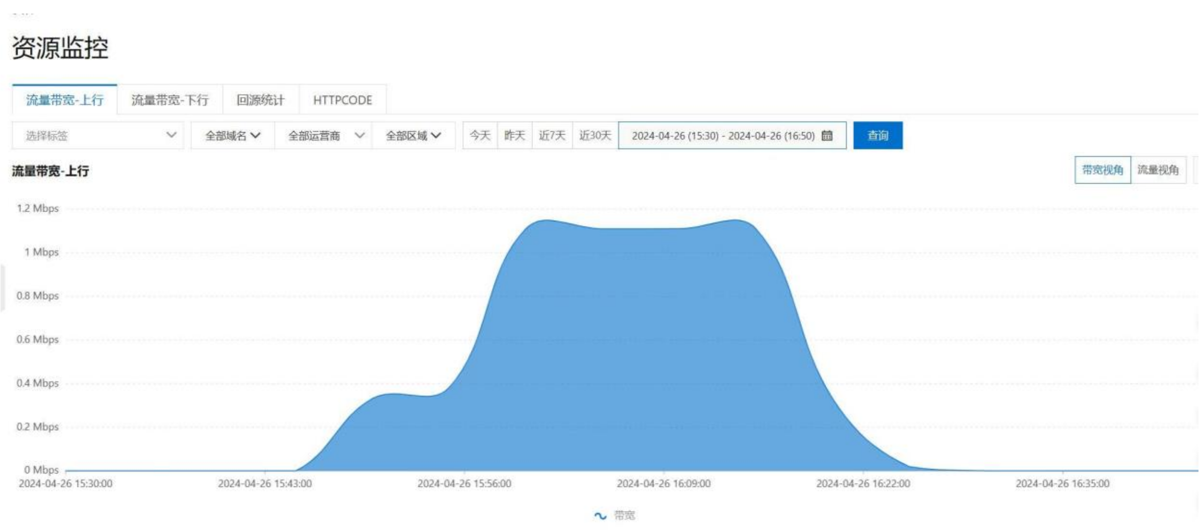

figure29. Live streaming data

Of course, we have also implemented the recording of live videos and cloud platform storage functions. When we choose to end the live broadcast, the cruise images will be automatically saved and uploaded to the OSS cloud platform for recording, similar to the "black box" of civil aviation aircraft.

3.3.4 4G sim card communication module

We have implemented a 4G module through a communication card, which is used to create a small local area network or directly transmit information to other terminals using 4G signals.

3.3.5 YOLO car recognition part

We adopted YOLOv5 and trained a model completely autonomously. The training process is as follows.

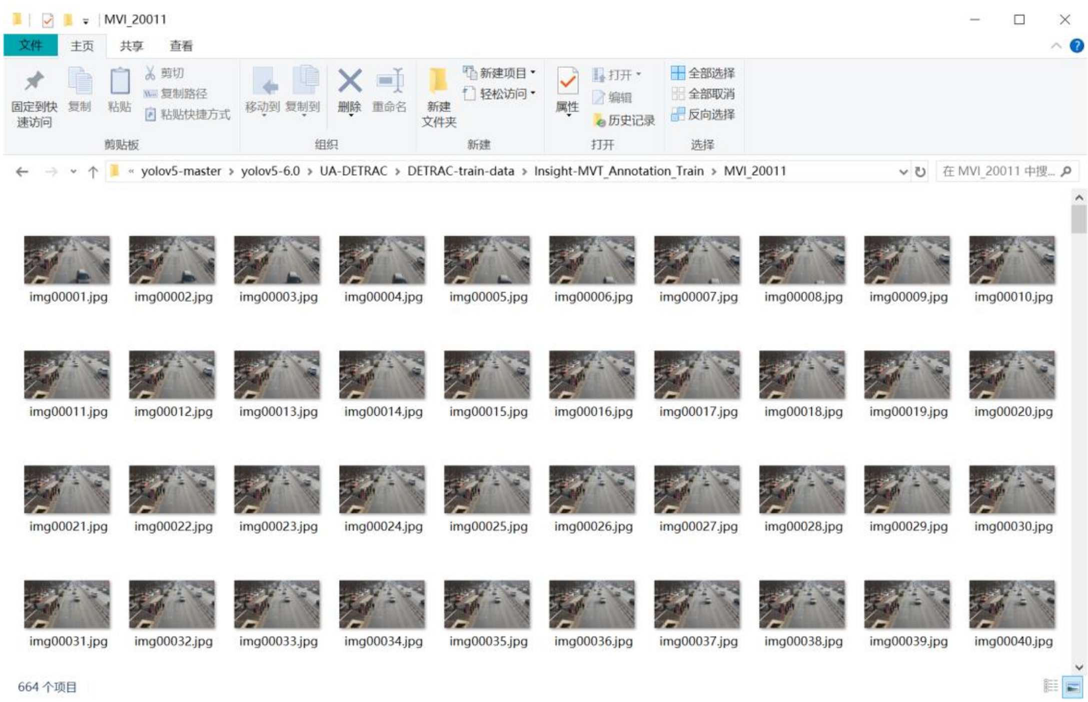

figure30. picture data atlas

figure31. Annotation file

e	train/b	train/o	train/cl	metrics/p	metrics	metrics	metrics/m	val/bo	val/ob	val/cls	x	x	x/
0	0.087937	0.049659	0.020328	0.85182	0.29447	0.36731	0.13417	0.071884	0.054159	0.01546	0.003332	0.003332	0.070014
1	0.062574	0.039983	0.013392	0.81748	0.60996	0.70846	0.37697	0.059641	0.045212	0.009071	0.006665	0.006665	0.040014
2	0.055072	0.035946	0.009561	0.89193	0.7701	0.87069	0.5641	0.053481	0.040436	0.006636	0.009998	0.009998	0.010013
3	0.050751	0.033327	0.007856	0.93203	0.84739	0.92611	0.6542	0.049007	0.036502	0.005237	0.009998	0.009998	0.009998
4	0.048366	0.031573	0.006952	0.94191	0.87019	0.94136	0.68805	0.04717	0.034599	0.004735	0.009998	0.009998	0.009998
5	0.047063	0.030694	0.00653	0.94863	0.88843	0.9508	0.7076	0.046183	0.033632	0.004473	0.009997	0.009997	0.009997
6	0.0463	0.030024	0.006239	0.95259	0.89614	0.95574	0.71952	0.045556	0.032984	0.004315	0.009995	0.009995	0.009995
7	0.045799	0.029654	0.006066	0.9572	0.89803	0.95805	0.72698	0.045139	0.032612	0.004218	0.009992	0.009992	0.009992
8	0.045342	0.029289	0.005886	0.95617	0.90337	0.95948	0.73111	0.044882	0.032339	0.004154	0.009989	0.009989	0.009989
9	0.045018	0.02905	0.005768	0.95787	0.90551	0.96081	0.73504	0.044697	0.032166	0.004106	0.009986	0.009986	0.009986
10	0.044657	0.028778	0.005663	0.95992	0.90654	0.96187	0.73833	0.044547	0.032015	0.004066	0.009982	0.009982	0.009982
11	0.044425	0.028626	0.005572	0.96019	0.90864	0.96306	0.74133	0.044409	0.031869	0.004034	0.009978	0.009978	0.009978
12	0.044206	0.02848	0.005504	0.96444	0.90751	0.96415	0.74367	0.044306	0.031749	0.004002	0.009974	0.009974	0.009974
13	0.044013	0.028283	0.005441	0.96631	0.90835	0.96518	0.74582	0.044213	0.031644	0.003973	0.009969	0.009969	0.009969
14	0.043903	0.028235	0.005375	0.96602	0.91076	0.96614	0.74897	0.044102	0.031524	0.003942	0.009963	0.009963	0.009963
15	0.043809	0.028179	0.005361	0.96559	0.91312	0.96712	0.75091	0.043985	0.031408	0.003915	0.009957	0.009957	0.009957
16	0.043616	0.027904	0.005319	0.96507	0.91615	0.96807	0.75348	0.043868	0.031281	0.003883	0.009951	0.009951	0.009951

figure32. Identification results

The YOLO version and personal PC features we use cannot perform real-time image recognition, resulting in significant transmission and processing delays. So we choose to recognize all the images during the recording process after the live video ends.

Our model has the basic ability to recognize cars through a large number of basic car image annotations and recognition, and can recognize cars in general environments.

4. Costs

4.1 Parts

Following is a starter table for parts costs.

TABLE 3

Table X Parts Costs

Part	Manufacturer	Retail Cost (RMB)	Bulk Purchase Cost (RMB)	Actual Cost (RMB)
EPO fuselage and wings	Making the fuselage	50	50	50
Motors and paddles	Driving an airplane	350	350	350
Battery	energy supply	160	160	160
Fixings (carbon tubes, screws, etc.)	stationary part	200	200	200
mc6c remote control	remote control	130	130	130
BLHe li-S ESCs	electronic speed controller	45*5=225	225	225
Pixhawk2.4.8	flight controller	400	400	400
Rasperrypi 3B	The main part of the entire visual system	350	350	350
4G module	Data transmission and communication	95	95	95
Ali Cloud server	Data storage and parallel computing	180	158.5	317
Website domain name*2	Push flow and pull flow	17	13.5	27
CSI Camera	Vedio capture	65	65	65
Total				

4.2 Labor

Labor cost: The labor cost is an important part for the senior design and the cost are estimated as below.The estimated salary for person is 100 ￥/ h (standard salary for Zhejiang University undergraduates). The normal work time per week is estimated for 40 hours (10 hours per person) according to our estimation for the senior design. We have 9 weeks to complete our senior design project.

$$\frac{40 hours}{week} * \frac{100 ¥}{hour} * 9 week = 36000 ¥$$

5. Conclusion

In the process of designing and building drones, we successfully constructed the basic structure and controlled the circuits, as well as achieved visual effects. The wing and fuselage design we adopted successfully reduced drag and increased load, and the perfect combination with the five axis circuit control system improved energy efficiency. In addition, in terms of visual design, we can not only access the video images captured by unmanned aerial vehicle cameras through any terminal and network, but also visually recognize and analyze the images. At present, we are able to recognize cars with the required accuracy. In the future, we hope to transmit the identified results in real time and back to the drone to achieve path planning.

5.1 Accomplishments

One of the standout achievements of this project was the successful integration and testing of an electric propulsion and control system. This system meets our initial performance metrics, providing enhanced thrust-to-weight ratios essential for the eVTOL's vertical takeoff and landing capabilities. The propulsion and control system demonstrated reliability across multiple test flights, establishing a solid foundation for future scalability.

In the process of visual development, we successfully solved the Raspberry PI startup, the use of push module and camera to complete the picture capture and automatic push stream, based on the Ari cloud server to complete the web broadcast and AI vehicle recognition, as well as large capacity flight record storage.

From the moment the plane starts, the Raspberry PI will be connected to the power supply, and then all the programs will be turned on in sequence in the self-start script, so that we can complete the complete process of transmitting the picture from the drone to the server for real-time broadcast and background recognition.

5.2 Uncertainties

In the process of visual development, we still encountered some problems, for example, the video broadcast built by Ali Cloud server cannot be directly placed on the live broadcast platform, but requires a certain video format transcoding to be viewed by normal users.

In addition, although the parallel computing power of the server has been very powerful, we still can not directly identify the good video in real time to the live stream, so we can only live broadcast and video recognition separately, and the identification of the good results stored in oss. Although we ended up using a number of methods to optimize our live streaming, there was still a lot of work to be done.

5.3 Ethical considerations

In strict adherence to the guiding principles and ethical standards outlined in the IEEE Code of Ethics, our organization has undertaken a comprehensive approach to preemptively tackle any potential ethical issues that might arise from the deployment of EVLOT drones within various contexts. Our due diligence encompasses stringent measures throughout different stages of drone involvement, which includes both testing and operational phases.

During the origination and execution of flight trials, along with the active use period of these unmanned aerial systems, we diligently consult and rigorously abide by the latest aviation regulations that govern airspace management and the piloting of drone technology. This ensures that our operations are not only legal but harmoniously integrated with the pre-existing frameworks that ensure the safety and coordination of multifaceted air traffic activities.

Our commitment extends to the incorporation of advanced technological safeguards within our EVLOT drone line. These drones have been meticulously designed to include state-of-the-art safety mechanisms. The obstacle avoidance feature has undergone extensive testing to confirm its reliability in preempting potential collisions during flight operations. Moreover, emergency response protocols have been refined to enable the drone to manage unexpected scenarios effectively. Another pivotal safety component is the automatic return-to-home function, which is activated in instances where the drone's control signals are compromised, ensuring the unmanned craft navigates back to its origin without manual intervention.

In pursuit of environmental stewardship, EVLOT drones have been engineered to operate via electric power, foregoing the reliance on fossil fuels and thus considerably reducing carbon emissions and other pollutants associated with traditional transportation methods. The structural utilization of foam panels substantially lessens the overall weight of the drone, thereby enhancing energy efficiency and further diminishing its environmental footprint.

The protection of user data and personal privacy stands paramount in our design philosophy. The cloud infrastructure, entrusted with the storage and transmission of sensitive information, employs robust encryption protocols. These cryptographic shields are vigorously tested against unauthorized access, guaranteeing alignment with prevailing privacy protection legislation. We conscientiously ensure that all personal data obtained through our drone operations is safeguarded with the utmost respect for confidentiality and integrity, thereby reinforcing the trust placed in us by our users.

Conclusively, the steps we have undertaken to fortify our EVLOT drones transcend mere regulatory compliance; they echo our unwavering dedication to upholding the highest ethical standards, securing the safety of our skies, preserving ecological balance, and protecting the invaluable trust of our customers.

In consideration of maintaining the highest standards of electrical safety, our team has taken a proactive approach by selecting a circuit board that integrates a sophisticated battery management system. This system is pivotal for monitoring and safeguarding the health of the batteries by ensuring proper charging and discharging cycles. Moreover, we have bolstered our safety measures with the implementation of overload protection mechanisms. These are instrumental in preventing potential damages from an excessive inflow or outflow of electrical current. Additionally, our comprehensive safety strategy includes a short circuit protection feature to preempt any risks posed by unintentional circuit bridgings, which can lead to severe malfunctions or even hazards.

Our dedication to safety extends beyond just electrical aspects and into structural considerations. For mechanical safety, our design conforms rigorously to established international or regional safety protocols. We regularly engage in thorough inspections and perform conscientious maintenance to guarantee that this adherence results in long-term reliability and safety. Furthermore, all structural parameters of the fuselage are meticulously calculated to match exacting specifications. Such precise engineering is complemented by an innovative emergency shutdown mechanism. This critical safety feature is devised to enable a rapid cessation of flight operations in exigent circumstances, thereby curtailing potential damage.

Taking lab safety to the next level, we leverage advanced simulation software to foresee and prepare for any operational scenarios the Unmanned Aerial Vehicle (UAV) might encounter in real-world conditions. This preemptive measure entails multiple rounds of virtual testing to mitigate any foreseeable risks before UAVs are physically employed. Our lab test regime escalates from static analyses to dynamic assessments within controlled environments, concluding with full-scale operation tests that are carried out in unenclosed spaces. To further ensure lab safety, our analysts have conducted exhaustive risk evaluations aimed at identifying, managing, and counteracting potential laboratory test hazards, fostering a secure working environment for our engineers and technicians.

Finally, when considering the end-user's safety, our approach is twofold. Firstly, we employ sophisticated software capable of imposing constraints on the UAV such as geofencing, which restricts operation within predefined geographical boundaries. Secondly, by limiting the flight altitude and range through software restrictions, we significantly reduce the incidence of accidents attributable to user error. These measures play a vital role in sustaining operational safety and integrity, providing users with peace of mind while handling our state-of-the-art UAV technology.

5.4 Future work

Although the eVTOL drone achieved reliable flight durations, there is room for improvement in its self-stabilize system. Future work will focus on developing more advanced self-stabilize technologies and algorithms. Additionally, extending flight times and reducing charging intervals are also aims we could enhance.

Going forward, we may be able to optimize our training model, and optimize our push stream transmission process, by adding transmission modules or using a more powerful single-chip microcomputer to optimize the overall process, in the hope of achieving true real-time identification live, rather than doing our two functions separately.

In addition, in order to make Alibaba Cloud's live stream can be watched by the audience stably at one time, we may need to do better web construction or directly produce a complete app to make these functions better integrated development.

References

[1] Introduction and figure of Mission Planner Vtol survey: https://ardupilot.org/planner/docs/mission-planner-overview.html

[1] Alexey Bochkovskiy, ChienYao Wang, and HongYuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. apr 2020.

[2] Jacob Culley, Sam Garlick, Enric Gil Esteller, Petar Georgiev, Ivan Fursa, Isaac Vander Sluis, Peter Ball, and Andrew Bradley.System design for a driverless autonomous racing vehicle. 2020 12th International Symposium on Comunication Systems, Networks and Digital Signal Processing (CSNDSP), pages 1-6, 2020.

[3] 李文胜.基于树莓派的嵌入式 Linux 开发教学探索[J].电子技术与软件工程,2014,(9):219-220.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5).

[5] Raspberry Pi 3 Model B. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-3-model-b/

[6] A. Ladaycia, A. Mokraoui, K. Abed-Meraim, and A. Belouchrani, “Performance bounds analysis for semi-blind channel estimation inmimo-ofdm communications systems,” IEEE Transactions on Wireless Communications, vol. 16, no. 9, pp. 5925 – 5938, Sept. 2017

[7] Design Reference from other product: http://www.vrtianxia.com/about/tool/html

[8] Assembly of drones: https://www.zhihu.com/question/310301226